Chapter 11

Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida

Matthew A. Jaro, System Automation Corporation

A test census of Tampa, Florida and an independent postenumeration survey (PES) were conducted by the U.S. Census Bureau in 1985. The PES was a stratified block sample with heavy emphasis placed on hard-to-count population groups. Matching the individuals in the census to the individuals in the PES is an important aspect of census coverage evaluation and consequently a very important process for any census adjustment operations that might be planned. For such an adjustment to be feasible, record-linkage software had to be developed that could perform matches with a high degree of accuracy and that was based on an underlying mathematical theory. A principal purpose of the PES was to provide an opportunity to evaluate the newly implemented record-linkage system and associated methodology. This article discusses the theoretical and practical issues encountered in conducting the matching operation and presents the results of that operation. A review of the theoretical background of the record-linkage problem provides a framework for discussions of the decision procedure, file blocking, and the independence assumption. The estimation of the parameters required by the decision procedure is an important aspect of the methodology, and the techniques presented provide a practical system that is easily implemented. The matching algorithm (discussed in detail) uses the linear sum assignment model to "pair" the records. The Tampa, Florida, matching methodology is described in the final sections of the article. Included in the discussion are the results of the matching itself, an independent clerical review of the matches and nonmatches, conclusions, problem areas, and future work required.

KEY WORDS: Census adjustment; Census coverage evaluation; EM algorithm; Postenumeration survey.

1. INTRODUCTION

Record-linkage methodology and software were developed at the U.S. Bureau of the Census during the past several years primarily to support census coverage evaluation efforts. By matching individuals counted in a census to those counted in an independent postenumeration (or pre-enumeration) survey, estimates of the quality of the enumeration can be produced. An important use of matching is to support an adjustment operation if it is decided to adjust the 1990 decennial census.

Clerical procedures typically used for such evaluations are too costly, unreproducible, error-prone, and time-consuming to be a viable alternative for such an adjustment (especially in view of the fact that state-level tabulations are due to the U.S. president by December 31, 1990). Therefore, the technical success of any adjustment procedure rests primarily on the ability to match a large number of records quickly, economically, and accurately. Even a few matching errors may be of critical importance, since population adjustments can be less than 1% in some instances. A complete discussion of the adjustment and census methodology issues can be found in Citro and Cohen (1985), Ericksen and Kadane (1985), and Wolter (1986).

Record linkage has numerous applications in both the private and public sectors. Examples include purging a list of duplicates, determining multiple-frame survey overlap, and geographic coding.

The Record Linkage Staff of the Statistical Research Division was established to implement a statistically justifiable, economical, and accurate record-linkage system to replace previous ad hoc systems and to reduce the number of cases that must be manually matched (see Jaro 1985).

Generalized computer programs have been written to implement the methodology discussed in this article. The first test of this software was the 1985 census of Tampa, Florida. The actual matching was conducted using a personal computer, although a mainframe version of the software also exists. Generalization is achieved through a program that automatically "writes" a customized program that will perform the matching for a particular application. The user specifies the fields to be matched, the record formats, parameters, blocking variables, etcetera, and the generation program creates a matcher that can be run with the desired files. This software generation technique results in a program that executes efficiently—a requirement for matching very large files.

This article presents the theoretical background necessary to understand the statistical basis of record linkage in general, the methodology developed for the estimation of parameters required by any record-linkage activity, the basic algorithmic approach used by the matcher, the specific methodology used for matching the 1985 census of Tampa to the postenumeration survey (PES), and the results of this process.

2. THEORETICAL CONCEPTS

2.1 Background

Consider two computer files, A and B, consisting of records taken from a population. Each file consists of a

* Matthew A. Jaro is Director of Research and Development, System Automation Corporation, Silver Spring, MD 20910. This work was accomplished while he was a Principal Researcher, Statistical Research Division, U.S. Bureau of the Census. The author acknowledges the contributions of R. P. Kelley on the parameter estimation methodology; Danny R. Childers, who designed and tabulated the PES; and Sue Finnegan, who directed the manual matching activities.

Journal of the American Statistical Association
June 1989, Vol. 84, No. 406, Applications and Case Studies

number of fields, or "components," and a number of records, or "observations." Typically, each observation corresponds to a member of the population and the fields are attributes identifying the individual observation, such as name, address, age, and sex. The objective of the record linkage or matching process is to identify and link the observations on each file that correspond to the same individual. The records are taken to contain no unique identifiers that would make the matching operation trivial. That is, the individual fields are all subject to error.

We can define two disjoint sets **M** and **U** formed from the cross-product of **A** with **B**, the set **A** × **B**. A record pair is a member of set **M**, if that pair represents a true match. Otherwise, it is a member of **U**. The record-linkage process attempts to classify each record pair as belonging to either **M** or **U**.

2.2 Weights

The components (fields) in common between the two files are useful for matching. Not all components, however, contain an equal amount of information, and error rates vary. For example, a field such as sex only has two value states and consequently could not impart enough information to identify a match uniquely. Conversely, a field such as surname imparts much more information, but it may frequently be reported or transcribed (keyed) incorrectly.

Weights are used to measure the contribution of each field to the probability of making an accurate classification. Newcombe and Kennedy (1962) discussed the concept of weights based on probabilities of chance agreement of component value states. Fellegi and Sunter (1969) extended these concepts into a more rigorous mathematical treatment of the record-linkage process. Their definition of weights takes into account the error probabilities for each field by using a log-likelihood ratio. Let m_i = Pr{component i agrees $\mid r \in \mathbf{M}$} and u_i = Pr{component i agrees $\mid r \in \mathbf{U}$} for all record pairs r. If, for a given record pair, component i agrees (matches), then the weight for component i, $w_i = \log_2(m_i/u_i)$. If component i disagrees, then the weight $w_i = \log_2((1 - m_i)/(1 - u_i))$.

2.3 Decision Procedure

For any record pair, a composite weight can be computed by summing the individual component weights. Since $m_i > u_i$ in most cases, fields that agree make a positive contribution to this sum, whereas fields that disagree make a negative contribution. A most significant concept advanced by Fellegi and Sunter (1969) is an optimal decision procedure for record linkage. For this procedure, three states are defined. A record pair is classified as a match if the composite weight is above a threshold value, a nonmatch if the composite weight is below another threshold value, and an undecided situation if the composite weight is between these two thresholds.

The threshold values can be calculated (see Sec. 3.4) given the acceptable probability of false matches (the probability that a record pair is classified as a match when the records do not represent the same individual) and the probability of false nonmatches.

2.4 Estimation of the u_i

Values for the m_i and the u_i probabilities must be estimated for each pair of files to be matched. Estimating the u_i (the probability that a component agrees given **U**) is simplified by the fact that the cardinality of the set **U** (denoted by $|\mathbf{U}|$) is generally much greater than that of **M**. For two files, both of equal size, F, $|\mathbf{M}| = pF$, where p is the proportion of matched pairs, and $|\mathbf{U}| = F^2 - pF$. Consequently, estimates for the u probabilities can be obtained by ignoring the contribution from **M** and considering only the probability of chance agreement of the component i. Usually this can be estimated from a sample of pairs rather than from all pairs.

Estimating the m_i probabilities (the probability that a component agrees given **M**) is more difficult. Conditioning on **M** presupposes an a priori knowledge of F correctly matched pairs. This could be obtained by a prelinked sample of the population. If such a sample were obtained clerically, much expense would be involved and the error rates for the clerical operation might be too high to permit accurate parameter estimation. One solution is blocking and using a latent trait model.

2.5 Blocking

For files of average size $|\mathbf{A} \times \mathbf{B}|$ is too great to consider all possible record pairs. Since there are many more record pairs in **U** than in **M** and 2^n possible comparison configurations involving n fields, drawing record pairs at random would require a sample size approaching all record pairs (for typical applications) to obtain sufficient information about the relatively rare **M** cases.

The two files can be partitioned into mutually exclusive and exhaustive blocks designed to increase the proportion of matched pairs observed while decreasing the number of record pairs to compare. Comparisons are restricted to record pairs within each block. Consequently, blocking is important for the actual matching and for parameter estimation activities. Blocking is generally implemented by means of sorting the two files on one or more variables. For example, if both files were sorted by zip code, the pairs to be compared would only be drawn from those records where zip codes agree. Record pairs disagreeing on zip code would not be considered and hence would be automatically classified as nonmatches (elements of **U**).

To be effective at enriching the **M** cases, such blocking variables must contain a large number of value states that are fairly uniformly distributed and such variables must have a low probability of reporting error (i.e., a high weight). Blocking is a trade-off between computation cost (examining too many record pairs) and false nonmatch rates (classifying record pairs as nonmatches because the records are not members of the same block). Multiple-pass matching techniques using independent blocking variables for each run can minimize the effect of errors in a set of blocking variables. R. P. Kelley has developed an algo-

rithm that may assist in choosing the best blocking scheme in light of these trade-offs (see Kelley 1984).

3. PARAMETER ESTIMATION METHODOLOGY

3.1 Comparison Configuration Frequencies

This section discusses the methodology used to estimate the m_i probabilities. Information about the comparison configurations observed is provided by the matching software itself, which tabulates frequencies for all 2^n possible patterns of agreement and disagreement on n fields. To increase the proportion of matched pairs examined, these tabulations are performed using just those record pairs in which both observations come from the same block. Fortunately, the m_i probabilities may reasonably be expected to be independent of the blocking schemes chosen as long as errors in the blocking variables do not exclude an excessive number of matched records from the tabulations. This is consistent, however, with the goal of choosing blocking variables with low reporting error rates. The independence of the m_i probabilities to choices in blocking and the effect of errors in the blocking variables to the final estimates are yet to be determined.

Given frequencies for all possible agreements and disagreements, the m_i probabilities can be estimated using any of several procedures. The EM algorithm described here is the most effective of those developed and tested.

3.2 The EM Algorithm

Given n fields and a sample of N record pairs drawn from $A \times B$, let $\gamma_i^j = 1$ if field i agrees for record pair j, let $\gamma_i^j = 0$ if field i disagrees for record pair j, for $i = 1, \ldots, n$ and $j = 1, \ldots, N$. Further, let γ^j be the vector of ones and zeros showing field agreements and disagreements for the jth pair in the sample, and let γ be the vector containing all of the γ^j.

The m_i and u_i probabilities can be defined as $m_i = \Pr\{\gamma_i^j = 1 \mid r_j \in \mathbf{M}\}$ and $u_i = \Pr\{\gamma_i^j = 1 \mid r_j \in \mathbf{U}\}$ for a randomly selected record pair r_j and $i = 1, 2, \ldots, n$. Define p as the proportion of matched pairs equal to $|\mathbf{M}|/|\mathbf{M} \cup \mathbf{U}|$. The elements of $\mathbf{M} \cup \mathbf{U}$ (i.e., all record pairs r_j) are distributed according to a finite mixture with the unknown parameters $\Phi = (\mathbf{m}, \mathbf{u}, p)$. We will use an EM algorithm to estimate these parameters; in particular, the $\mathbf{m}$ vector is of the greatest interest. Notation is consistent with that used in Dempster, Laird, and Rubin (1977).

Let $\mathbf{x}$ be the complete data vector equal to $\langle \gamma, g \rangle$, where $g_j = (1, 0)$ iff $r_j \in \mathbf{M}$ and $g_j = (0, 1)$ iff $r_j \in \mathbf{U}$. Then, the complete data log-likelihood is

$$\ln f(\mathbf{x} \mid \Phi) = \sum_{j=1}^{N} g_j \cdot (\ln \Pr\{\gamma^j \mid \mathbf{M}\}, \ln \Pr\{\gamma^j \mid \mathbf{U}\})^T + \sum_{j=1}^{N} g_j \cdot (\ln p, \ln(1 - p))^T.$$

Although it is quite reasonable to expect the empirical frequencies to belie independence, since errors in one component will often induce errors in another, such departures will not likely disturb the ordering by composite weights of the 2^n configurations. Consequently, we will assume an independence model:

$$\Pr(\gamma^j \mid \mathbf{M}) = \prod_{i=1}^{n} m_i^{\gamma_i^j}(1 - m_i)^{1-\gamma_i^j} \quad (1)$$

and

$$\Pr(\gamma^j \mid \mathbf{U}) = \prod_{i=1}^{n} u_i^{\gamma_i^j}(1 - u_i)^{1-\gamma_i^j}. \quad (2)$$

Our application of the EM algorithm begins with estimates of the unknown parameters ($\hat{\mathbf{m}}$, $\hat{\mathbf{u}}$, $\hat{p}$) and consists of iterative applications of the expectation (E) and maximization (M) steps until the desired precision is obtained. The algorithm is not particularly sensitive to starting values, and the initial $\hat{m}$ values can be guessed. It is important, however, that the $\hat{m}$ values be greater than their corresponding $\hat{u}$ values. For Tampa, .9 was used for all of the initial $\hat{m}$. Estimation of $\hat{u}$ is discussed in Section 2.4.

For the E step, replace g_j with $(\hat{g}_m(\gamma^j), \hat{g}_u(\gamma^j))$, where

$$\hat{g}_m(\gamma^j) = \frac{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j}(1 - \hat{m}_i)^{1-\gamma_i^j}}{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j}(1 - \hat{m}_i)^{1-\gamma_i^j} + (1 - \hat{p}) \prod_{i=1}^{n} \hat{u}_i^{\gamma_i^j}(1 - \hat{u}_i)^{1-\gamma_i^j}}.$$

$\hat{g}_u(\gamma^j)$ can be derived similarly.

For the M step, the complete data log-likelihood can be separated into three maximization problems. Setting the partial derivatives equal to 0 and solving for $\hat{m}_i$ yields

$$\hat{m}_i = \sum_{j=1}^{N} [\hat{g}_m(\gamma^j)\gamma_i^j] \Big/ \sum_{j=1}^{N} [\hat{g}_m(\gamma^j)].$$

Further, the matrix of second partial derivatives can be shown to be negative-definite.

In practice, we store frequency counts, $f(\gamma^j)$ for each of the possible 2^n patterns of γ^j. These counts are obtained as follows: each file is partitioned into blocks by means of the blocking variables. For each block, all record pairs in the block are examined. For each record pair, the comparison vector γ^j is computed and 1 is added to the frequency count for that particular configuration. There are 2^n such counters. The counters are not reset after a block is processed, but represent the number of observations of each configuration over all blocks. Both estimation and actual matching are accomplished using the same blocks. The EM algorithm is run once using these frequencies.

The E step computes $\hat{g}_m(\gamma^j)$ for each of the 2^n patterns. This can be done without examining the individual observations, since the frequencies are a sufficient statistic for the M step. By replacing the individual observations with the frequencies, we obtain

$$\hat{m}_i = \sum_{j=1}^{2^n} [\hat{g}_m(\gamma^j)\gamma_i^j f(\gamma^j)] \Big/ \sum_{j=1}^{2^n} [\hat{g}_m(\gamma^j) f(\gamma^j)].$$

The arguments for the $\mathbf{u}$ probabilities are similar.

Finally, the proportion of matched pairs p can be esti-

mated by

$$\hat{p} = \sum_{j=1}^{N} [\hat{g}_m(\gamma^j)]/N = \sum_{j=1}^{2^n} [\hat{g}_m(\gamma^j) f(\gamma^j)] \Big/ \sum_{i=1}^{2^n} f(\gamma^j).$$

It must be remembered that the frequencies used for the p, **m**, and **u** estimates were obtained from record pairs within blocks and represent an accumulation over all blocks. Since blocking greatly reduces the number of nonmatched pairs observed and because blocking selects record pairs that are likely to match, the **u** probability estimates obtained using blocked data will be biased. Consequently, the **u** probabilities must be computed directly on unblocked data, as explained in Section 2.4, and the EM algorithm is only used to compute the **m** probabilities, where blocking enriches the number of matched pairs observed while avoiding comparisons on relatively large numbers of unmatched pairs.

The EM algorithm is highly stable and the least sensitive to the starting values of any of the methods studied. The algorithm is very simple to implement, and the probabilities will always be within bounds. The other methods are based on numerical analysis techniques, and it is possible for probabilities to exceed 1. The greater stability of the EM algorithm comes from the fact that logarithms lower the degree of the equations, whereas the method-of-moments techniques described subsequently use squared products of probabilities that are close to 1 and 0.

Comparison of the convergence criteria, rapidity of convergence, and sensitivity to independence for these methods are currently being studied.

3.3 Other Estimation Methods

The second estimation technique studied involves minimizing a system of 2^n equations (one for each comparison vector configuration) using the IMSL routine ZXSSQ [minimum of the sum of squares of m functions in n variables using a finite difference Levenberg–Marquardt algorithm (see IMSL 1984)]. The system was more sensitive to initial values, and it was possible for solution sets to have probabilities out of bounds.

The third method examined was due to Fellegi and Sunter (1969, app. II). The authors presented an algebraic solution for three fields, but it is easy to generalize the equations and solve the system of nonlinear simultaneous equations using numerical methods. The results agree with the other two methods used. However, the system is rather sensitive to the starting values, and in one case a penalty function had to be introduced to keep the probabilities within bounds.

3.4 Calculation of Threshold Values

An algorithm in our matcher determines the threshold weights as follows. There are 2^n possible configurations of agreement and disagreement of n components. These configurations can be ordered by the composite weight (the sum of the individual weights, w_i, for each component). After ordering the composite weights, the sum of Pr(• | **M**) and 1 minus the sum of Pr(• | **U**) can be calculated [see Eqs. (1) and (2), Sec. 3.2]. The maximum weight for a nonmatch decision is the weight of the configuration where the sum of Pr(• | **M**) does not exceed the desired probability that a matched pair should be classified as unmatched. The minimum weight for a match decision is the weight of the configuration where 1 minus the sum of Pr(• | **U**) does not exceed the desired probability that an unmatched pair should be classified as matched. Weights between these two thresholds are undecided cases.

For applications such as census matching, with approximately 10 components, this technique is computationally feasible. Unpublished experimentation has been performed by sampling component configurations for problems having many components, but the large number of cells makes it difficult to obtain a sufficient number of observations in each cell, so sampling error is not an overpowering factor.

4. MATCHING ALGORITHM

This section describes the basic operation of the matcher. Before matching, fields such as house address should be separated into components and spellings should be standardized. Both files must be sorted by the blocking variables.

4.1 Composite Weight Calculation

The matcher processes one block at a time, building a matrix (C) containing the composite weights for all pairs within the block being processed. The composite weights are computed by summing the individual weights for agreement or disagreement on each field (see Sec. 2.2). The simple agreement/disagreement dichotomy modeled by the theory is too simplistic for noncategorical fields. For example, character strings are compared using an information–theoretic character comparison algorithm that provides for random insertion, deletion, replacement, and transposition of characters. The weight assigned for such comparisons is prorated according to a measure of similarity between character fields (see Jaro 1978, pp. 106–108). If two character fields match exactly, the full weight for agreement is assigned to the comparison. If they disagree slightly, however, it would be wrong to assign the disagreement weight. Consequently, the weight assigned for the comparison will be somewhat less than the full agreement weight.

Similarly, weights for integer or continuous variables such as age can be prorated proportionally to the ratio of the difference and the minimum of the two values being compared (delta percent). For example, if age disagrees by one year in an 80-year-old man, it is less serious a mismatch than for a 1-year-old baby.

4.2 Assignment

After the matrix containing the composite weights for all pairs within the block is constructed (C_{ij} in the following), the records can be paired up (assigned). One record on file A can be assigned to one and only one record on file B, and vice versa. We wish to choose an assignment

scheme that maximizes the sum of the composite weights of the assigned record pairs. This is a degenerate transportation problem known as the linear sum assignment problem, which can be solved by a simple method requiring only addition and subtraction. The use of such a linear programming model to provide the assignments represents an advance over previous ad hoc assignment methods. The problem can be formulated as follows: Maximize

$$Z = \sum_{i=1}^{k} \sum_{j=1}^{k} C_{ij} X_{ij}$$

subject to

$$\sum_{j=1}^{k} X_{ij} = 1, \qquad i = 1, 2, \ldots, k,$$

and

$$\sum_{i=1}^{k} X_{ij} = 1, \qquad j = 1, 2, \ldots, k,$$

where C_{ij} is the cost (weight) of matching record i on file A with record j on file B, X_{ij} is an indicator variable that is 1 if record i is assigned to record j and 0 if i is not assigned to j, k_a is the number of records in the block being processed from file A, k_b is the number of records in the block being processed from file B, and k = maximum(k_a, k_b). If $k_a \neq k_b$, the matrix is made square (with dimension k) by inserting entries whose values are large negative numbers (less than any possible composite weight). This prevents these entries from being assigned.

An excellent discussion of the theory of assignment problems can be found in Cooper and Steinberg (1974, chap. 11). The computer algorithm was obtained from Burkard and Derigs (1981) and is highly efficient and economical of storage since the original matrix elements remain unaltered (the computations are performed on vectors, since all operations apply to entire rows or columns).

Once an optimal assignment vector is obtained, an assigned pair can be classified as a match if the composite weight is greater than the Fellegi–Sunter threshold value. After all assigned pairs in the block are processed, the records for the next block can be read.

4.3 Duplicates

Duplicates can be detected by examining each row or column of the assignment matrix. If more than one entry is above the cutoff threshold, then there is a possibility of a duplicate. Two similar records on both files would probably be two separate individuals (a father and son, for example), but two similar records on only one file would probably be a duplicate.

4.4 File Preparation

To match any file, free-form information must be standardized. This is especially true of fields such as street address and person name. The components of the name should be separated into individual fields (given name, middle initial, and surname). This is much more effective and accurate than trying to match an entire name as a single character string. For street address, the various components of the address should be placed in individual fields and the spellings of common abbreviations (such as BD, BLVD) should be standardized. Punctuation should be removed from the fields.

The technique of SOUNDEX encoding (Knuth 1973, pp. 391–392) is a method of transforming a person's name into some code that tends to bring together all variants of the same name. For example, *Smith* and *Smythe* would both be coded as S530. Surname is often an important blocking variable. To maximize the chance that similarly spelled surnames reside in the same block, the SOUNDEX system can be used to code the names, and the SOUNDEX code can be used as a blocking variable. There are better encoding schemes than SOUNDEX, but SOUNDEX with relatively few states and poor discrimination helps ensure that misspelled names receive the same code.

SOUNDEX is not recommended for matching nonblocking variables, since nonphonetic errors result in different codes and different names may receive the same code.

5. TAMPA MATCHING METHODOLOGY

This section describes the computer match of the 1985 census of Tampa, Florida, to the PES. The object of the matching study was to identify all individuals who responded to both the PES and to the census. The records consisted of individual data and contained name, address, and demographic characteristics. The primary goal of the computer matcher was to eliminate the first-level clerical match (where matches could be determined with relatively unsophisticated personnel). The system exceeded this goal. A multiple blocking strategy was used to increase the numbers of matched records given errors in the blocking variables. The strategies are called Pass I and Pass II, respectively.

5.1 Pass I Match

The following variables were used for matching:

1. Census block numbering area (CBNA) (blocking variable)
2. Census block number (blocking variable)
3. Surname (SOUNDEX) (blocking variable)
4. Given name (m = .98, u = .09)
5. Middle initial (m = .35, u = .03)
6. Relation to head of household (m = .39, u = .20)
7. Sex and marital status (combined) (m = .82, u = .21)
8. Birthdate (m = .94, u = .04)
9. Race and Hispanic origin (combined) (m = .90, u = .67)
10. Street name (m = .96, u = .03)
11. House number (m = .99, u = .01)
12. Apartment number (m = .35, u = .26)

The blocking variables for Pass I were census block numbering area (CBNA), census block number, and SOUN-

DEX code of surname. CBNA and census block number were used as blocking variables, since only census data for PES sample blocks were keyed and, consequently, it would be unlikely that data would be available for units geocoded to incorrect blocks. All records failing to match in Pass I would participate in the Pass II match, which used different variables for blocking.

The results of the Pass I match were as follows: 7,358 PES records read, 8,798 census records read, 4,375 matched pairs, 165 nonclassified pairs, 628 unmatched PES records, 702 unmatched census records, 2,190 skipped PES records, and 3,556 skipped census records. Records are said to be *skipped* when one or more of the blocking variables do not match. The nonclassified, unmatched, and skipped records are input to the Pass II process.

5.2 Pass II Match

In an attempt to match records that failed to match in Pass I, an independent blocking scheme was chosen for Pass II. The blocking variables were CBNA, census block number, SOUNDEX of street name, house number, and apartment number. CBNA and block number were reused, since no records exist outside of the sample area and detecting geocoding errors would be unlikely.

Apartment number had to be used, since some high-rise developments contained more than 500 units at a single address and the matcher has a maximum block size that cannot be exceeded. Subsequently, the matcher was modified to correct this problem by flagging such "overflow" blocks, which could be processed in a separate subsequent run.

A blank apartment number may mean either that the apartment number is not appropriate or that the value was not reported. Since apartment numbers are sometimes not appropriate, blank apartment numbers would be accepted as a valid value.

The Pass II match was useful, since it displayed record pairs and groups in household sequence.

The results from Pass II were as follows: 2,983 PES records read, 4,423 census records read, 212 matched pairs, 885 nonclassified pairs, 1,114 unmatched PES records, 1,321 unmatched census records, 772 skipped PES records, and 2,005 skipped census records. The matching decisions were made very conservatively to limit the number of false matches. This is important where estimation of relatively rare events is required (such as for undercount estimation). The tight error tolerances account for the high number of nonclassified cases. Most of these could be resolved quickly, since records are already paired. All nonclassified pairs, unmatched records, and skipped records would be processed clerically. Many records were unmatched because of construction and demolition in the PES area, vacancies, noninterviews, proxy data, geocoding errors, etcetera.

The total number of records matched automatically from both passes was 4,587, with 885 nonclassified pairs that could be rapidly resolved. Approximately 40 minutes (wall-clock time) were required to conduct the Pass I match on an IBM PC-AT, with 20 minutes required for Pass II. Sorts required about 15 minutes each.

6. CLERICAL REVIEW AND FINAL MATCHING RESULTS

After the computer matching was completed, the records were grouped by household and printed on a computer-generated matching form for clerical review. Many of the nonmatches were easily converted to matches by reviewing the persons in the household together.

A total of 5,343 persons were matched in the entire process, with 4,587 matched by the computer (85.9%). Of the 885 nonclassified persons, 225 were at vacant addresses or were noninterviews, leaving 660 persons that could be resolved clerically. Of these 660 persons, 83.39% were determined to be actual matches. The number of persons who were either matched automatically or with a quick verification of the computer-assigned possible match is 5,177 (computed by 4,587 persons matched automatically plus 83.39% of 660 persons). A total of 5,177 out of the 5,343 cases yield an effective match rate of 96.89% for the automated system, leaving only about 3% of the cases for extensive clerical intervention. The clerical and professional review staffs were able to match only 19.47% of the residual nonmatched records from the computer operation.

6.1 Review of Computer Matches

All of the matches assigned by the computer were reviewed to evaluate the computer matching. Eight persons assigned by the computer matcher were actual errors, yielding an error rate of .174% (8 of 4,587). Inferences regarding the matcher's accuracy, however, should not be made from this one case study, as results can vary with the accuracy of the geographic reference material, data entry procedures, and numbers of hard-to-count population groups in a specific area.

6.2 Matching in Neighboring Blocks and Duplicates

The census questionnaire information was only entered for the PES sample blocks. Therefore, it was impossible to detect geographic coding errors automatically by the computer (since no machine-readable data were available). The blocks that bordered the sample blocks, however, were searched clerically to attempt to reduce the number of nonmatches.

Of 1,692 persons not matched in the sample blocks, 726 were matched to neighboring blocks, resulting in a 42.9% reduction in the nonmatch rate. The largest reduction was in blocks that were predominantly black and Hispanic and contained multi-unit structures.

The matching processes (both automatic and clerical) include the detection of duplicate records for an individual. Searching on neighboring blocks uncovered 32 census duplicates. The total number of duplicates within the sample and surrounding blocks was 145 (2.6% of the matched records).

7. CONCLUSIONS AND FUTURE WORK

The automated matching system greatly exceeded expectations in terms of both match rate and accuracy: 96.89% of the records that were matched were matched either automatically or could be quickly verified. The error rate was .174% (again, no inferences should be drawn from this). The 1986 test census of Los Angeles, California, includes an automated extended search to detect movers and geocoding errors. Several matching errors were due to problems in high-rise multi-unit structures. A new methodology that should eliminate these problems has been developed.

The use of the EM algorithm for parameter estimation appears to be the most promising of all techniques attempted in terms of insensitivity to choice of starting values and ease of implementation.

Additional work is required in a number of areas. Questions to be answered include: What is the sensitivity of the final classification to parameter estimation error and statistical dependence of reporting errors and/or value states of fields? Can successful models for multiple state comparison vectors be developed? For example, instead of agreement and disagreement states, the vector could be augmented to include cases where values are missing (currently, these receive zero weight). Can cutoff thresholds be computed by means of a closed-form equation without enumerating 2^n configurations, or can such a form be developed for changes in only one weight? If so, then weights could be adjusted by means of a Bayesian procedure to account for changes in the distribution of the value states of a field in different geographic areas, and, further, weights could be adjusted where value state distributions are likely to be skewed. For example, an agreement on a name like Humperdinck should carry more weight than an agreement on Smith, but strictly speaking, the cutoff thresholds would change if the weights for a field were changed, and a closed-form equation for the thresholds would permit changing the cutoff values for particular cases during the matching process. Can the errors in both the automated and manual phases of matching be properly modeled, and can variances be computed?

A calibration data set is being developed from the Tampa, Florida, experience. A linked file such as this can be used to measure sensitivity and the relative merits of various matching schemes. I will attempt to answer systematically most of the questions posed in this article and to improve the mathematical underpinning of record-linkage methodology.

[*Received January 1987. Revised October 1988.*]

REFERENCES

Burkard, R. E., and Derigs, U. (1981), "Assignment and Matching Problems: Solution Methods With FORTRAN-Programs," in *Lecture Notes in Economics and Mathematical Systems* (No. 184), New York: Springer-Verlag, pp. 1–11.

Citro, C. F., and Cohen, M. L. (1985), *The Bicentennial Census, New Directions for Methodology in 1990*, Washington, DC: National Academy Press.

Cooper, L., and Steinberg, D. (1974), *Methods and Applications of Linear Programming*, Philadelphia: W. B. Saunders.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood From Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society*, 39, 1–38.

Ericksen, E. P., and Kadane, J. P. (1985), "Estimating the Population in a Census Year: 1980 and Beyond" (with discussion), *Journal of the American Statistical Association*, 80, 98–131.

Fellegi, I. P., and Sunter, A. B. (1969), "A Theory for Record Linkage," *Journal of the American Statistical Association*, 64, 1183–1210.

International Mathematical and Statistical Libraries, Inc. (1984), *User's Manual*, Houston: Author.

Jaro, M. A. (1978), "UNIMATCH: A Record Linkage System, User's Manual," Washington, DC: U.S. Bureau of the Census.

——— (1985), "Current Record Linkage Research," in *Proceedings of the Statistical Computing Section, American Statistical Association*, pp. 140–143.

Kelley, R. P. (1984), "Blocking Considerations for Record Linkage Under Conditions of Uncertainty," in *Proceedings of the Social Statistics Section, American Statistical Association*, pp. 602–605.

Knuth, D. E. (1973), *The Art of Computer Programming, Volume 3: Sorting and Searching*, Reading, MA: Addison-Wesley.

Newcombe, H. B., and Kennedy, J. M. (1962), "Record Linkage," *Communications of the Association for Computing Machinery*, 5, 563–566.

Wolter, K. M. (1986), "Some Coverage Error Models for Census Data," *Journal of the American Statistical Association*, 81, 338–346.

Matt Jaro, Founder, President & CEO:

MatchWare Technologies Inc.'s founder, CEO and director of technology, Matt Jaro, enjoys 30+ years of experience in the science of probabilistic record linkage and information technology. Jaro has led MatchWare to a position of global leadership in the practical implementation and use of this methodology.

Matt Jaro conceived, designed, and authored MatchWare's proprietary software. Well known as a speaker and author in the fields of address matching and geographic information systems, Matt is regarded as an international authority on probabilistic record linkage methodology.

Jaro holds degrees in mathematics from California State, and computer science from George Washington. Prior to founding MatchWare, he held a variety of information technology positions with public and private sector organizations including: Booz-Allen Applied Research, the U.S. Census Bureau, The Corporation for Applied Systems, Public Technology, Inc., and System Automation.

In the mid-80's, Jaro was a principal researcher at the U.S. Census Bureau where he developed the mathematical methodology and software to perform statistically valid matching procedures in support of estimating census coverage. Although application specific, the census estimation methodology Matt developed was precedent-setting in the field of probabilistic record linkage.